Prof. Dr. Gerd Stumme, Robert Jäschke
Fachgebiet Wissensverarbeitung

23.1.2008

10. Übung „Künstliche Intelligenz“

Wintersemester 2007/2008

Beschreibungslogik

1. Betrachten Sie die folgende Ontologie:

   - $Animal \sqsubseteq Thing$
   - $Plant \sqsubseteq Thing$
   - $Animal \sqcap Plant \equiv \bot$
   - $domain(eats) = Animal$
   - $Herbivore \equiv \forall eats.Plant$
   - $Carnivore \equiv \forall eats.Animal$

   a) Erklären Sie, warum ein Reasoner schließt, daß *Plant* eine Unterklasse von *Herbivore* und *Carnivore* ist.

   > „It is because Plants can't eat anything (they are disjoint from Animal, and the domain of eats is Animal), so they trivially satisfy the condition that everything they eat is both Animal and Plant, so they are both Carnivores and Herbivores. Using OilEd or Protege with either FaCT or Racer gives this inference.“
   >
   > *Ian Horrocks*

   Aus der Aussage, daß die Domain von *eats* Animal ist, kann also zunächst geschlossen werden, daß eine Pflanze nichts ißt. Rufen wir uns die Semantik von $\forall R.C$ in Erinnerung:

   $$(Herbivore \equiv \forall eats.Plant)^{\mathcal{I}}$$

   $$Herbivore^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b : (a, b) \in eats^{\mathcal{I}} \Longrightarrow b \in Plant^{\mathcal{I}}\}$$

   $$Herbivore^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b : a\, eats^{\mathcal{I}}\, b \Longrightarrow b \in Plant^{\mathcal{I}}\}$$

   Da für jedes $a \in Plant^{\mathcal{I}}$ kein b existiert mit $a\, eats^{\mathcal{I}}\, b$, ist die Implikation in der Klammer stets wahr und damit sind alle Pflanzen Pflanzenfresser. Auf gleiche Weise kann man schließen, daß alle Pflanzen Fleischfresser sind.

b) Dies scheint nicht vernünftig zu sein. Wie würden Sie die Ontologie verbessern, um das Problem zu beheben?

> Es hilft, wenn man Herbivore und Carnivore stattdessen folgendermaßen definiert:
> - $Herbivore \equiv \forall eats.Plant \sqcap Animal$
> - $Carnivore \equiv \forall eats.Animal \sqcap Animal$

2. Es soll nun das Konzept „vegetarische Pizza“ definiert werden. Welche der folgenden Definitionen ist dafür angemessen? Geben Sie dazu jeweils eine natürlichsprachliche Beschreibung der logischen Formeln an.

a) $VegetarischePizza \equiv Pizza \sqcap \neg\exists hatBelag.(Fleisch \sqcap Fisch)$

> Eine vegetarische Pizza ist eine Pizza die keinen Belag hat, der gleichzeitig Fleisch und Fisch ist.

b) $VegetarischePizza \equiv Pizza \sqcap \forall hatBelag.(\neg Fleisch \sqcup \neg Fisch)$

> Eine vegetarische Pizza ist eine Pizza und wenn diese einen Belag hat, so ist dieser kein Fleisch oder kein Fisch.

c) $VegetarischePizza \equiv Pizza \sqcap \neg\exists hatBelag.Fleisch \sqcap \neg\exists hatBelag.Fisch$

> Eine vegetarische Pizza ist eine Pizza die keinen Belag aus Fisch und keinen Belag aus Fleisch hat.

d) $VegetarischePizza \equiv Pizza \sqcap \exists hatBelag.\neg Fleisch \sqcap \exists hatBelag.\neg Fisch$

> Eine vegetarische Pizza ist eine Pizza die einen Belag hat der nicht Fleisch ist und einen Belag hat der nicht Fisch ist.

e) $VegetarischePizza \equiv Pizza \sqcap \forall hatZutat.(\neg Fleisch \sqcap \neg Fisch)$

> Eine vegetarische Pizza ist eine Pizza deren Zutaten weder Fleisch noch Fisch enthalten.

> Die Definition 2e schließt aus, daß die Pizza Fleisch oder Fisch enthält und ist daher am besten geeignet, eine vegetarische Pizza zu definieren.
> Geht man davon aus, daß der Boden einer Pizza stets vegetarisch ist, so ist auch die Definition 2c geeignet, eine vegetarische Pizza zu beschreiben.

3. Sei R eine Rolle. Welche Eigenschaft von R wird durch $(R^{\mathcal{I}})^+ = R^{\mathcal{I}}$ definiert?

> Hat R diese Eigenschaft, so ist es eine *transitive* Rolle.

4. Testen Sie mittels des Tableauverfahrens die Erfüllbarkeit von

$$\forall hatZutat.(\exists hatZutat.\text{Salz}) \sqcap \exists hatBelag.Fleisch \sqcap \forall hatBelag.(\neg Fleisch \sqcup \neg \text{Salz}) \sqcap \exists hatZutat.\text{Salz},$$

wobei *hatZutat* eine transitive Rolle ist.

Der besseren Schreib- und Lesbarkeit wegen werden folgende Abkürzungen verwendet:

S: Salz

F: Fleisch

Z: hatZutat

B: hatBelag

Die zu bearbeitende Formel lautet also:

$$\forall Z.(\exists Z.S) \sqcap \exists B.F \sqcap \forall B.(\neg F \sqcup \neg S) \sqcap \exists Z.S.$$

Wir starten mit der Anwendung der $\sqcap$-Regel und erhalten somit den Knoten w zusammen mit der Menge $\mathcal{L}(w)$:

$\mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\}$

w

Durch Anwendung der $\exists$-Regel erhalten wir den Knoten x, welcher mit w durch eine mit der Rolle B beschrifteten Kante verbunden ist:

$\mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\}$

$$\begin{tikzcd} & w \arrow[dl, "B"'] \\ x & \end{tikzcd}$$

$\mathcal{L}(x) = \{F\}$

Durch Anwenden der $\forall$-Regel fügen wir weitere Elemente zu $\mathcal{L}(x)$ hinzu:

$$\begin{tikzcd} & \mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\} \\ & w \arrow[dl, "B"'] \\ \mathcal{L}(x) = \{F, \neg F \sqcup \neg S\}\ x & \end{tikzcd}$$

Wir können die $\sqcup$-Regel auf $\mathcal{L}(x)$ anwenden und $\neg F$ hinzufügen:

$$\begin{tikzcd} & \mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\} \\ & w \arrow[dl, "B"'] \\ \mathcal{L}(x) = \{F, \neg F \sqcup \neg S, \neg F\}\ x & \end{tikzcd}$$

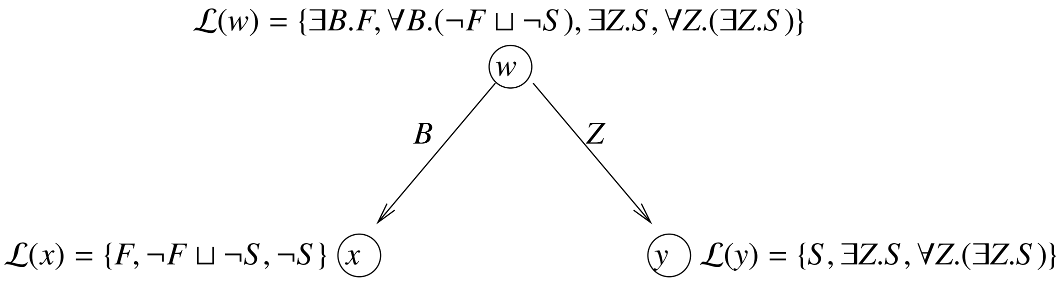
Dies führt zu einem Clash, denn F und $\neg F$ sind unerfüllbar.
Also fügen wir stattdessen $\neg S$ zu $\mathcal{L}(x)$ hinzu:

$$\begin{tikzcd} & \mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\} \\ & w \arrow[dl, "B"'] \\ \mathcal{L}(x) = \{F, \neg F \sqcup \neg S, \neg S\}\ x & \end{tikzcd}$$

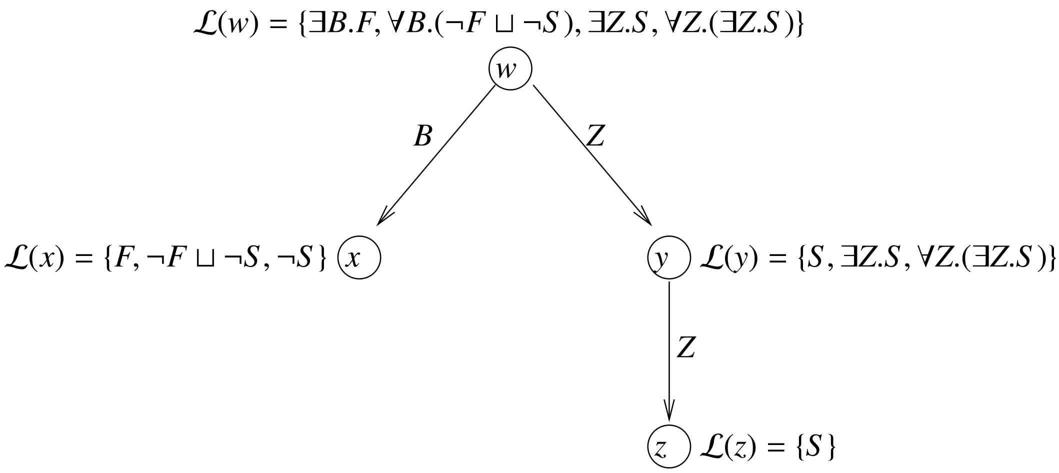
Jetzt wenden wir die $\exists$-Regel nochmals auf $\mathcal{L}(w)$ an und erhalten den Knoten y und eine mit der Rolle Z beschriftete Kante von w nach y:

$$\begin{tikzcd} & \mathcal{L}(w) = \{\exists B.F, \forall B.(\neg F \sqcup \neg S), \exists Z.S, \forall Z.(\exists Z.S)\} & \\ & w \arrow[dl, "B"'] \arrow[dr, "Z"] & \\ \mathcal{L}(x) = \{F, \neg F \sqcup \neg S, \neg S\}\ x & & y\ \mathcal{L}(y) = \{S\} \end{tikzcd}$$

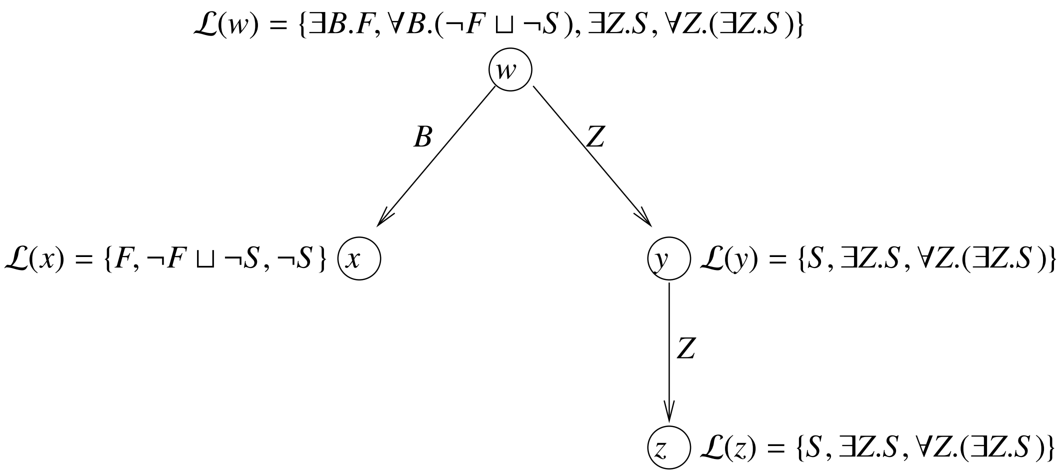
Durch die $\forall$-Regel fügen wir weitere Elemente zu $\mathcal{L}(y)$ hinzu. Hier kommt zum Tragen, dass Z eine transitive Rolle ist, denn deswegen müssen wir nicht nur $\exists Z.S$ sondern auch $\forall Z.(\exists Z.S)$ zu $\mathcal{L}(y)$ hinzufügen:



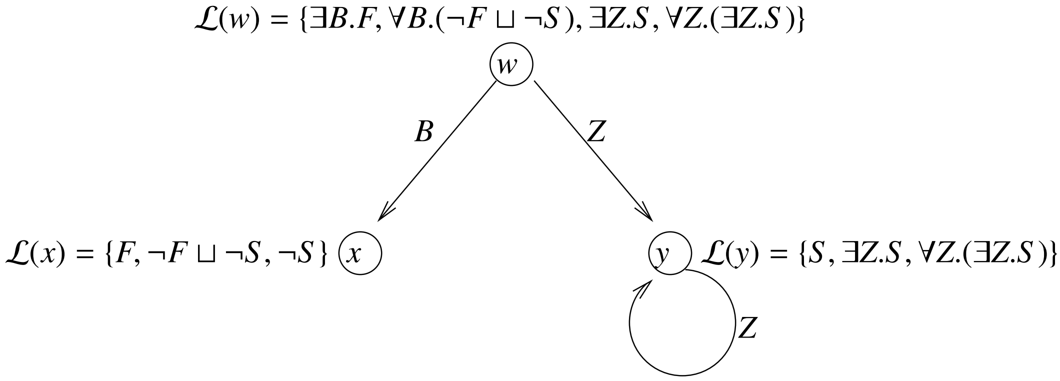
Wenden wir auf $\mathcal{L}(y)$ die $\exists$-Regel an, erhalten wir einen weiteren Knoten z zusammen mit einer Kante von y nach z, welche mit der Rolle Z beschriftet ist:



Anwenden der $\forall$-Regel auf $\mathcal{L}(y)$ erweitert die Menge $\mathcal{L}(z)$, so dass sie gleich $\mathcal{L}(y)$ ist:



Der Knoten z hat damit einen Vorgängerknoten y mit $\mathcal{L}(z) \subseteq \mathcal{L}(y)$ und wird daher geblockt:



Damit endet das Tableau-Verfahren mit einem vollständigen, Clash-freien Baum. Die Wissensbasis ist daher erfüllbar.